Chess: Dissected
Group: 8-23

Chia Yu Zhe Ashton (5)
Chong Choon-Hou Rafael (7)
Chu Yu Heng Ian (8)
Chan Yi Hong Josiah (13)

Class: 1O1

Written Report

Introduction

Chess has recently been gaining popularity. We are all chess players, with two of our group members playing chess competitively (Ian Chu and Ashton Chia). Therefore, we decided to research chess and develop a better understanding of the math behind a game of chess. Chess is

also a very complex game and therefore we like to understand it better from a Mathematics perspective.

Literature Review

White, John C., "A MATHEMATICAL ANALYSIS OF THE GAME OF CHESS" (2018). Selected Honors Theses. 101. "A MATHEMATICAL ANALYSIS OF THE GAME OF CHESS" by John C. White

This thesis explains how chess can be played out using probability, statistics etc. From these math skills, a chess player's skills can be significantly improved however, not perfect.

Zermelo's work

Zermelo brought up two questions. Firstly, when a player is in a "winning" position can this be defined in a mathematical sense. Secondly, if a player is in a "winning" position, can we determine the number of moves needed to force a win?To answer the first question, Zermelo came to the conclusion that a player should at least theoretically force a win or a draw. To answer the second question, Although there is no solid answer as every game of chess is played differently which leads to the number of moves in order to win every game different, there is a limit to the number of moves needed to win. The limit is that the number of moves would not exceed the number of positions in the game. In conclusion, a player's chess skills cannot be perfect but by using mathematical skills, we are definitely able to at least force a draw.

Brown, Alfred James, "Knight's Tours and Zeta Functions" (2017). Master's Theses. 4836. DOI: https://doi.org/10.31979/etd.e7ra-46ny https://scholarworks.sjsu.edu/etd_theses/4836

This thesis explores in what dimensions a Knight's Tour is possible and impossible using Graph Theory and investigating the Knight's Tour on a Hamiltonian Path.

Research Question 1:
What is the math behind a Knight's tour?

Ans: Graph Theory

Using the Graph Theory, we can have a better understanding of how the knight's tour can be solved. A knight's tour occurs when the knight can move to each of the remaining 63 squares. The knight only visits each square once. There are two types of knight's tours. The first type is a closed tour (Hamiltonian cycle) where the knight's tour can repeat itself (the last square **can** reach the starting square). The second type is an open tour (Non-Hamiltonian cycle) where the knight's tour cannot repeat (the ending square **cannot** reach the starting square). A Hamiltonian Cycle is when there is a closed-loop on a graph where every point is visited exactly once. A loop is just an edge that joins a point to itself, so a Hamiltonian cycle is a path travelling from a point back to itself, visiting every point en route.

Graph theory is a graph with a collection of points (vertices) connected by lines (edges), relating some relationship between the points).

Example:

The vertex that represents the a1 square would have edges connecting it to the vertices represented by b3 and c2 only since those are the only two squares the knight can jump to from a1.
From a more central square, say e4, the vertice representing e4 would have edges connecting it to 8 other vertices, namely those represented by: f2, g3, g5, f6, d6, c5, c3 and d2.

From all this information, we can create a board where each square is represented by a dot, and each dot is linked by a line in a manner in which a knight moves, this would show every possible Knight tour:

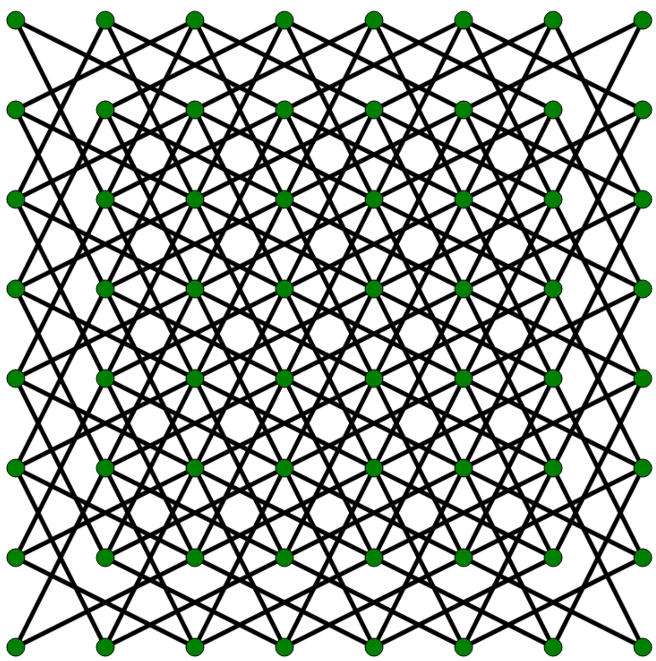


Image Source: *https://upload.wikimedia.org/wikipedia/commons/9/91/Knight%27s_graph.sv*g

We can calculate the exact number of Closed Knight's Tours by finding the number of squares a knight can go to from each different square through this graph.
8 Legal edges= 16 Different vertices
7 Legal edges= 0 Different vertices
6 Legal edges= 16 Different vertices
5 Legal edges= 0 Different vertices
4 Legal edges= 20 Different vertices

3 Legal edges= 8 Different vertices

2 Legal edges= 4 Different vertices

1 Legal edges= 0 Different vertices

Research question 2:

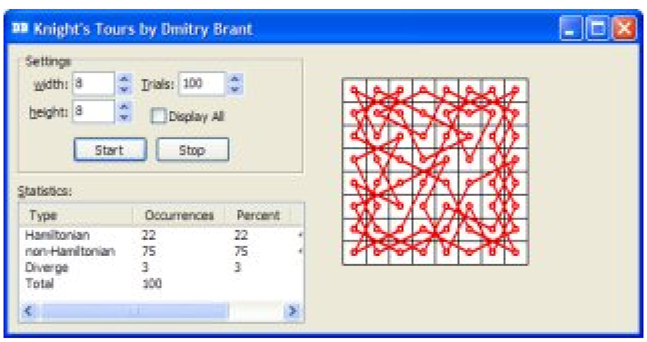
When is a closed Knight's Tour possible/impossible on an m x n and n x n chessboard?

Ans: When the **length and breadth** of the chessboard is greater than 4.

And the **total number of squares** in the chessboard is an even number

Dmitry Brant (https://dmitrybrant.com/knights-tour)

The code makes use of the neural network to solve the problem.



By inputting the number of trials to conduct, you can find the number of knight's tours possible in the chessboard.**Warnsdorff's Heuristic**

We can use Warnsdorff's rule which is a heuristic for finding a single knight's tour. The knight is moved to always proceed to the square from which the knight will have the *fewest* onward moves.

When calculating the number of onward moves for each candidate square, we do not count moves that revisit any square already visited. It is possible to have two or more choices for which the number of onward moves is equal; there are various methods for breaking such ties.

This rule may also more generally be applied to any graph. In graph-theoretic terms, each move is made to the adjacent vertex with the least degree. Although the Hamiltonian path problem is NP-hard in general, this heuristic can successfully locate a solution in linear time on many graphs that occur in practice. The knight's tour is such a special case.

Byron Phung (https://github.com/ByronPhung/knights-tour)

The code is a knight's tour solver/visualiser using Warnsdorff's heuristic.
The program prompts you for the location of the knight, on an 8x8 chessboard
The output of the code would look like this :

```
Enter x & y (1-8) separated by a space (e.g. 4 4): 6 5 (entered by you)

The knight's tour ended prematurely at (5,1) during move #60.

34 31 10  0 26 29  8  5
11  0 33 30  9  6 23 28
32 35  0 25 48 27  4  7
 0 12 59 36 53 24 49 22
60 43 54 47 58 37 18  3
13 46 57 52  1 50 21 38
42 55 44 15 40 19  2 17
45 14 41 56 51 16 39 20
```

10,000 trials will be conducted for each sized board. (Note : $n > 6$ and $m > 5$, $m \neq n$)

m x n	Number of closed knight's tours, Percentage	n x n	Number of closed knight's tours, Percentage
5 x 6	325, 3%	6 x 6	2802, 28%
5 x 7	0, 0%	7 x 7	0, 0%
6 x 7	3441, 34%	8 x 8	2657, 27%

5 x 8	3955, 40%	9 x 9	0, 0%
6 x 8	3642, 36%	10 x 10	1592, 16%
7 x 8	2548, 25%	11 x 11	0, 0%
		12 x 12	970, 10%

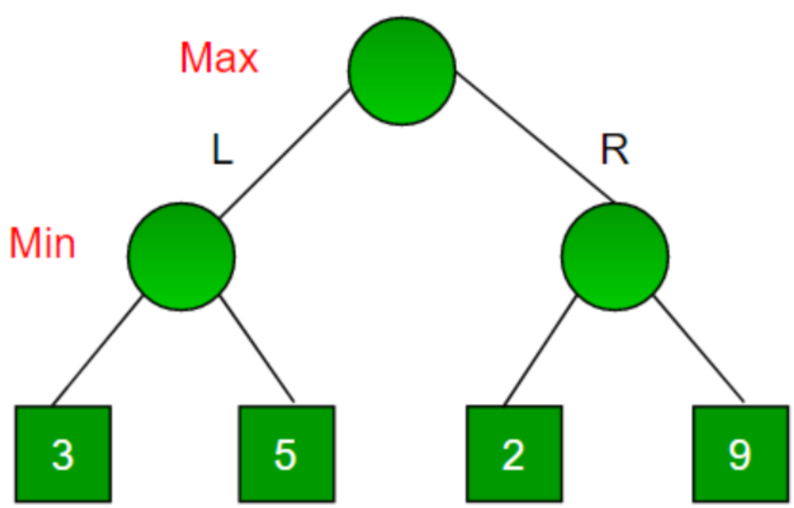
Research question 3:

How does a Chess Computer evaluate a position to decide on the best possible move in a position?

Ans: Minimax Algorithm

Firstly, chess computers can employ the *Minimax Algorithm* which is a decision based algorithm used in game theory. The computer would be the maximizer who will try to get the highest score possible while the minimizer will try to do the opposite and get the lowest score possible. (minimizer in this case would be player 2) If the maximizer has the upper hand then, the score of the board will tend to be some positive value. If the minimizer has the upper hand in that board state, it will have some negative value. So in this case, the maximizer and the minimizer can be represented as the white and black players respectively.

- **Example:**



Let us presume the Maximizer goes LEFT: It is now the minimizers turn. The minimizer now has a choice between 3 and 5. Being the minimizer it will conclude, choose the least among both, that is 3. It does so as it would minimize the losses, so this can be shown in a game of chess where the minimizer (player 2) would choose to lose a Knight (3 points) rather than losing a rook (5 points) this would help minimize the losses in the position.

Therefore, a chess computer when given these two choices will pick left as it wants to maximize its winnings, plus it knows what the minimizer's best move is. The computer will search and evaluate these trees that will go on and on depending on the computer's power, these trees would represent sequences of moves from the current position and the computer would attempt to execute the best such sequence during play/maximize winnings.

Conclusion

After completing the respective research above, other possible interesting topics that we could research include, but not limited to; The Eight Queens Puzzle, this was proposed by Max Bezzel in Germany in 1848, the eight queens puzzle is the problem of placing eight chess queens on an 8×8 chessboard so that no two queens threaten each other, this is a very interesting puzzle/problem as there are only 92 distinct solutions, this would definitely be a problem that we could explore. Geometry in Chess is also something we could research, for example, in some King and pawn endgames, without actually calculating various lines, it is possible to form an imaginary square around the pawn if the opposition's king is within said square, it is a draw, if it is not, then the player wins. Such simple geometric rules will definitely be interesting to explore in a game of chess, which would show the beautiful complexity of the relationship between mathematics and chess.

Reflection:

The research done in this project was hard and several obstacles were faced along the way, for example, answering research question 2 was a hard one as programming and coding was needed in order to achieve results, fortunately, one of our members was skilled enough to comprehend the code and we achieved the results we desired. Another example of an obstacle faced was the understanding of the research done on question 1, it was particularly difficult due our limited knowledge in terms of graph theory and the other stuff involved in answering this question. Lastly, condensing the information of the Minimax Algorithm, followed by applying that to a paragraph written in terms of a real chess game, was a challenging task as it required an understanding of both the game of chess and our research. Overall, researching and completing this project as a whole group was a very enriching and new experience that every one of us enjoyed.

References

Aradhya., A. (2021, March 31). Minimax algorithm in-game THEORY: Set 1 (Introduction). Retrieved April 04, 2021, from https://www.geeksforgeeks.org/minimax-algorithm-in-game-theory-set-1-introduction/

White, John C., "A MATHEMATICAL ANALYSIS OF THE GAME OF CHESS" (2018). Selected Honors Theses. 101., from https://firescholars.seu.edu/honors/101

Brown, A. J. (2017 July). Knight's Tours and Zeta Functions. Retrieved from https://scholarworks.sjsu.edu/cgi/viewcontent.cgi?article=8383&context=etd_theses

Squirrel, Douglas; Cull, P. (1996). "A Warnsdorff-Rule Algorithm for Knight's Tours on Square Boards" (PDF). Retrieved 2011-08-21.

I. Parberry. Scalability of a neural network for the knight's tour problem. Neurocomputing, 12:19-34, 1996.

Y. Takefuji and K. C. Lee. Neural network computing for knight's tour problems. Neurocomputing, 4(5):249-254, 1992.

GitHub - ByronPhung/knights-tour: Solves knight's tours using Warndorff's Rules and Python. GitHub. Retrieved August 15, 2021, from https://github.com/ByronPhung/knights-tour

Knight's Tours Using a Neural Network – Dmitry Brant. (n.d.). Https://Dmitrybrant.Com/Knights-Tour. Retrieved August 15, 2021, from https://dmitrybrant.com/knights-tour